

Microsoft DevOps Solutions: Automating Communication

IMPLEMENTING TRACEABILITY TO ENABLE
COMMUNICATION



Chris B. Behrens

SOFTWARE ARCHITECT

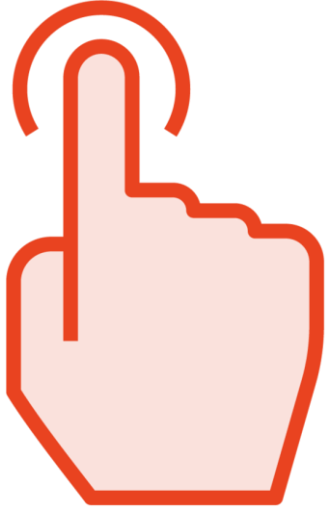
@chrisbbehrens



Keep communication
actionable.



Brevity Is the Soul of Wit



Communicate your
message **ONCE**



We got hundreds of
emails



In that sea of
communication, an
important alert was
missed

Communicate your
message ONCE in context.



Understanding Application Insights

An aspect of Azure Monitor

We'll cover Monitor later on



What Is Application Insights, Exactly?



A telemetry SDK

Examples:

- A user just requested a page from this address, and it took this long.
- An application made an Ajax call, and this many bytes were transferred.
- A request resulted in an unhandled exception, and here are the details

You can customize with your own code

How Application Insights Works

Server-side in Azure

Client-side (if the optional
Javascript is configured)

Identified by an
instrumentation key

And connected to our app
instance



Implementing AppCenter Support in Your Mobile Applications



Just Like Application Insights



Another SDK that you install and provide an instrumentation key to



Just like with App Insights, you can write custom code to take control of events



This will probably all be Application Insights someday



For now, we set up an export that pushes the data to App Insights



Two Key Terms

- **Crashes**
- **Analytics**



Nuts and Bolts

```
using Microsoft.AppCenter;  
using Microsoft.AppCenter.Analytics;  
using Microsoft.AppCenter.Crashes;
```



Get Yourself Connected

MainActivity.cs

```
protected override void OnCreate(Bundle savedInstanceState)
{
    AppCenter.Start("{YOUR INSTRUMENTATION KEY}", typeof(Analytics), typeof(Crashes));

    ...
}
```

We pass the two types, Analytics and Crashes

There are other interesting packages, like push and data

Setting This Up



Now, your data is reaching App Center

But we want to see all this in a single place

1. Add your Azure Account to App Center

2. Go to Export under Settings

3. Create the export

This will create a new app-specific App Insights instance

Easy



Get Yourself Connected

MainActivity.cs

```
protected override void onCreate(Bundle savedInstanceState)
{
    AppCenter.Start("{YOUR INSTRUMENTATION KEY}", typeof(Analytics), typeof(Crashes));

    ...
}
```

Secrets do not belong in
version control!



Connecting Azure DevOps to Third Party Tools



Service Connections

An API connection
between Azure
and a third-party

Facilitated
through a
Marketplace plug-
in

SonarCloud – one
of my favorites



Service Hooks

A simpler approach via an event subscription

The third party trusts Azure, then Azure authenticates



How Service Hooks Work

Provide

Provide auth token

Specify

Specify general event

Filter

Filter the event with hook-specific details



Connecting Your Communication in Azure DevOps



Root Cause Analysis in Build Engineering



If our build is finding problems

Then the more connected it is to other parts of your process, the better

1. Build => Commit

2. Commit => Developer

3. Commit => Work Item



Driving Builds off of Version Control

**Trigger your builds from
commits (or pull requests)**

**This connects your build to
the commit (step 1)**



Connecting Work Items to Commits



- 1. Branch is associated with a work item**
 - All child commits will be attached to the work item, too
- 2. Note the work item id in the commit message**
 - But you have to remember to do this
- 3. Clean it up after the fact by adding a link**
 - This will connect even existing builds to the work item